

Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №4
з навчальної дисципліни «Методи наукових досліджень»

Тема:

ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ ПРИ
ВИКОРИСТАННІ РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ ЕФЕКТУ
ВЗАЄМОДІЇ.

Виконав:
Студент групи ІВ-92,
Карпека Дмитрій Юрійович

Перевірив:
Регіда П. Г.

Київ 2021

Мета роботи

Провести повний трьохфакторний експеримент. Знайти рівняння регресії адекватне об'єкту.

Завдання на лабораторну роботу

1. Скласти матрицю планування для повного трьохфакторного експерименту.
2. Провести експеримент, повторивши N раз досліди у всіх точках факторного простору і знайти значення відгуку Y. Знайти значення Y шляхом моделювання випадкових чисел у певному діапазоні відповідно варіанту. Варіанти вибираються за номером в списку в журналі викладача.

$$y_{i\max} = 200 + x_{cp\max}$$

$$y_{i\min} = 200 + x_{cp\min}$$

$$\text{де } x_{cp\max} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}, \quad x_{cp\min} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$$

3. Знайти коефіцієнти рівняння регресії і записати його.
4. Провести 3 статистичні перевірки – за критеріями Кохрена, Стьюдента, Фішера.
5. Зробити висновки по адекватності регресії та значимості окремих коефіцієнтів і записати скореговане рівняння регресії.
6. Написати комп'ютерну програму, яка усе це моделює.

Варіант:

210	-20	15	-30	45	-30	-15
-----	-----	----	-----	----	-----	-----

Виконання роботи:

- 1) Результати роботи програми:

Матриця повного факторного експерименту з натуралізованими значеннями:

X0	X1	X2	X3	X12	X13	X23	X123	Y1	Y2	Y3	Aver Y	S_y
1	-20	-30	-30	600	600	900	-18000	180	214	185	193.0	224.67
1	-20	-30	-15	600	300	450	-9000	193	210	213	205.33	77.56
1	-20	45	-30	-900	600	-1350	27000	177	187	178	180.67	20.22
1	-20	45	-15	-900	300	-675	13500	209	211	190	203.33	89.56
1	15	-30	-30	-450	-450	900	13500	211	207	179	199.0	202.67
1	15	-30	-15	-450	-225	450	6750	211	201	211	207.67	22.22
1	15	45	-30	675	-450	-1350	-20250	209	209	188	202.0	98.0
1	15	45	-15	675	-225	-675	-10125	188	185	194	189.0	14.0

Матриця повного факторного експерименту з нормалізованими значеннями:

X0	X1	X2	X3	X12	X13	X23	X123	Y1	Y2	Y3	Aver Y	S_y
1	-1	-1	-1	1	1	1	-1	180	214	185	193.0	224.67
1	-1	-1	1	1	-1	-1	1	193	210	213	205.33	77.56
1	-1	1	-1	-1	1	-1	1	177	187	178	180.67	20.22
1	-1	1	1	-1	-1	1	-1	209	211	190	203.33	89.56
1	1	-1	-1	-1	-1	1	1	211	207	179	199.0	202.67
1	1	-1	1	-1	1	-1	-1	211	201	211	207.67	22.22
1	1	1	-1	1	-1	-1	-1	209	209	188	202.0	98.0
1	1	1	1	1	1	1	1	188	185	194	189.0	14.0

Рівняння

$$y = 197.5 + 1.92 * x_1 + -3.75 * x_2 + 3.83 * x_3 + -0.17 * x_1x_2 + -4.92 * x_1x_3 + -1.42 * x_2x_3 + -4.0 * x_1x_2x_3$$

Критерій Кохрена

Тест Кохрена: $G = 0.3$

Дисперсії однорідні з імовірністю 0.95

Критерій Ст'юдента

Коефіцієнт b_1 незначимий, тому виключається із рівняння регресії

Коефіцієнт b_2 незначимий, тому виключається із рівняння регресії

Коефіцієнт b_3 незначимий, тому виключається із рівняння регресії

Коефіцієнт b_5 незначимий, тому виключається із рівняння регресії

Коефіцієнт b_7 незначимий, тому виключається із рівняння регресії

Скореговане рівняння регресії:

$$y = 197.5 + 0 * x_1 + 0 * x_2 + 0 * x_3 + -0.17 * x_1x_2 + 0 * x_1x_3 + -1.42 * x_2x_3 + 0 * x_1x_2x_3$$

Критерій Фішера

Математична модель неадекватна експериментальним даним на рівні значимості 0.05.

Повтор експерименту для $m+1$

Матриця повного факторного експерименту з натуралізованими значеннями:

X0	X1	X2	X3	X12	X13	X23	X123	Y1	Y2	Y3	Y4	Aver Y	S_y
1	-20	-30	-30	600	600	900	-18000	211	213	187	207	204.5	106.75
1	-20	-30	-15	600	300	450	-9000	205	176	195	186	190.5	115.25
1	-20	45	-30	-900	600	-1350	27000	182	188	189	173	183.0	40.5

1	-20	45	-30	-900	600	-1350	27000	182	188	189	173	183.0	40.5	
1	-20	45	-15	-900	300	-675	13500	202	192	196	210	200.0	46.0	
1	15	-30	-30	-450	-450	900	13500	202	178	201	208	197.25	130.69	
1	15	-30	-15	-450	-225	450	6750	214	176	206	179	193.75	273.19	
1	15	45	-30	675	-450	-1350	-20250	189	198	207	212	201.5	77.25	
1	15	45	-15	675	-225	-675	-10125	205	200	185	183	193.25	89.19	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+														

Матриця повного факторного експерименту з нормалізованими значеннями:

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+														
X0	X1	X2	X3	X12	X13	X23	X123	Y1	Y2	Y3	Y4	Aver Y	S_y	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+														
1	-1	-1	-1	1	1	1	-1	211	213	187	207	204.5	106.75	
1	-1	-1	1	1	-1	-1	1	205	176	195	186	190.5	115.25	
1	-1	1	-1	-1	1	-1	1	182	188	189	173	183.0	40.5	
1	-1	1	1	-1	-1	1	-1	202	192	196	210	200.0	46.0	
1	1	-1	-1	-1	-1	1	1	202	178	201	208	197.25	130.69	
1	1	-1	1	-1	1	-1	-1	214	176	206	179	193.75	273.19	
1	1	1	-1	1	-1	-1	-1	189	198	207	212	201.5	77.25	
1	1	1	1	1	1	1	1	205	200	185	183	193.25	89.19	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+														

Рівняння

$$y = 195.47 + 0.97 * x1 + -1.03 * x2 + -1.09 * x3 + 1.97 * x1x2 + -1.84 * x1x3 + 3.28 * x2x3 + -4.47 * x1x2x3$$

Критерій Кохрена

Тест Кохрена: Gr = 0.311

Дисперсії однорідні з імовірністю 0.95

Критерій Ст'юдента

Критерій Ст'юдента

Коефіцієнт b2 незначимий, тому виключається із рівняння регресії

Коефіцієнт b4 незначимий, тому виключається із рівняння регресії

Коефіцієнт b6 незначимий, тому виключається із рівняння регресії

Коефіцієнт b7 незначимий, тому виключається із рівняння регресії

Скореговане рівняння регресії:

$$y = 195.47 + 0.97 * x1 + 0 * x2 + -1.09 * x3 + 0 * x1x2 + -1.84 * x1x3 + 0 * x2x3 + 0 * x1x2x3$$

Критерій Фішера

Математична модель адекватна експериментальним даним на рівні значущості 0.05

Код програми:

```
from random import randint
import prettytable
import math
from scipy.stats import f, t
```

```

from functools import partial

# Лабораторна робота №4 "ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ
# ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ З УРАХУВАННЯМ ЕФЕКТУ ВЗАЄМОДІЇ" з предмету МОПЕ
# Варіант №210 Карпека Дмитрій

#-----Початкові умови-----
variant = 210

min_x = [-20, -30, -30]
max_x = [15, 45, -15]

x0 = [1]
norm_x = [[-1, -1, -1],
           [-1, -1, 1],
           [-1, 1, -1],
           [-1, 1, 1],
           [1, -1, -1],
           [1, -1, 1],
           [1, 1, -1],
           [1, 1, 1]]
natur_x = [ [min_x[0], min_x[1], min_x[2]],
             [min_x[0], min_x[1], max_x[2]],
             [min_x[0], max_x[1], min_x[2]],
             [min_x[0], max_x[1], max_x[2]],
             [max_x[0], min_x[1], min_x[2]],
             [max_x[0], min_x[1], max_x[2]],
             [max_x[0], max_x[1], min_x[2]],
             [max_x[0], max_x[1], max_x[2]] ]

def experiment(m=3, n=8):

    regression_str = 'y = {} + {} * x1 + {} * x2 + {} * x3 + {} * x1x2 + {} *
x1x3 + {} * x2x3 + {} * x1x2x3'

    def matrix_plan(m, ymin, ymax, n):
        return [[randint(ymin, ymax) for _ in range(m)] for _ in range(n)]

    def multiplication(a, b):
        return a * b

    def average(list):
        return sum(list) / len(list)

    def round_to_2(number):
        return round(number, 2)

    def dispersion(list_y, aver_list_y):
        return [round_to_2(average(list(map(lambda y: (y - aver_list_y[i]) ** 2,
list_y[i])))) for i in range(len(list_y)))]

    def cochrane_criteria(S_y):
        global m
        print("\nКритерій Кохрена\n")
        Gp = max(S_y) / sum(S_y)
        q = 0.05
        q_ = q / f2
        chr = f.ppf(q=1 - q_, dfn=f1, dfd=(f2 - 1) * f1)

```

```

Gt = chr / (chr + f2 - 1)
print("Тест Кохрена: Gr = " + str(round(Gp, 3)))
if Gp < Gt:
    print("Дисперсії однорідні з імовірністю 0.95")
    pass
else:
    print("\nДисперсії неоднорідні.\nПовтор експерименту для m + 1\n")
    m = m + 1
    experiment(m)

def student_criteria(S_y, d):
    print("\nКритерій Ст'юдента\n")
    bettaList = [sum(S_y) * x0[0] / n,
                  average(list(map(multiplication, S_y, x1i))),
                  average(list(map(multiplication, S_y, x2i))),
                  average(list(map(multiplication, S_y, x3i))),
                  average(list(map(multiplication, S_y, norm_x12))),
                  average(list(map(multiplication, S_y, norm_x13))),
                  average(list(map(multiplication, S_y, norm_x23))),
                  average(list(map(multiplication, S_y, norm_x123)))]
    bettaList = [round_to_2(i) for i in bettaList]

    list_t = [bettaList[i] * S for i in range(n)]

    for i in range(n):
        if list_t[i] < t.ppf(q=0.975, df=f3):
            list_b[i] = 0
            d -= 1
            print('Коефіцієнт b' + str(i) + ' незначимий, тому виключається із рівняння регресії')
    print("\nСкореговане рівняння регресії:")
    print(regression_str.format(*map(round_to_2, list_b)))

def fisher_criteria(d):
    global m
    print("\nКритерій Фішера\n")
    f4 = n - d
    S_ad = (m * sum(
        [(list_b[0] + list_b[1] * x1i[i] + list_b[2] * x2i[i] + list_b[3] *
         x3i[i] + list_b[4] * norm_x12[i] +
         list_b[5] * norm_x13[i] + list_b[6] * norm_x23[i] + list_b[7] *
         norm_x123[i]
         - average_list_y[i]) ** 2 for i in range(n)]) / f4)
    Fp = S_ad / Sb

    if Fp > f.ppf(q=0.95, dfn=f4, dfd=f3): # перевірка критерію Фішера з використанням scipy
        print('Математична модель неадекватна експериментальним даним на рівні значимості 0.05.\nПовтор експерименту для m+1')
        m = m + 1
        experiment(m)
    else:
        print('Математична модель адекватна експериментальним даним на рівні значущості 0.05')

def printed_matrixes():
    pt1 = prettytable.PrettyTable()
    pt2 = prettytable.PrettyTable()
    pt1.field_names = ["X0", "X1", "X2", "X3"] + ["X12", "X13", "X23",

```

```

"X123"] + ["Y" + str(x) for x in range(1, m + 1)] + ["Aver Y"] + ["S_y"]
    pt2.field_names = ["X0", "X1", "X2", "X3"] + ["X12", "X13", "X23",
"X123"] + ["Y" + str(x) for x in range(1, m + 1)] + ["Aver Y"] + ["S_y"]

    print("Матриця повного факторного експерименту з натуралізованими
значеннями:\n")
    pt1.add_rows([x0 + natur_x[i] + natur_x12[i] + natur_x13[i] +
natur_x23[i] + natur_x123[i] + matrix_y[i] + [average_list_y[i]] + [S_y[i]] for
i in range(n)])
    print(pt1)

    print("\nМатриця повного факторного експерименту з нормалізованими
значеннями:\n")
    pt2.add_rows([x0 + norm_x[i] + [norm_x12[i]] + [norm_x13[i]] +
[norm_x23[i]] + [norm_x123[i]] + matrix_y[i] + [average_list_y[i]] + [S_y[i]]
for i in range(n)])
    print(pt2)

m = m
n = 8

x_average_max = sum(max_x) / 3
x_average_min = sum(min_x) / 3

y_max = round(200 + x_average_max)
y_min = round(200 + x_average_min)

matrix_y = matrix_plan(m, y_min, y_max, n)
average_list_y = [round(average(matrix_y[i]), 2) for i in
range(len(matrix_y))]

S_y = dispersion(matrix_y, average_list_y)

f1 = m - 1
f2 = n
f3 = f1 * f2
d = 4

Sb = sum(S_y) / n
S = math.sqrt(Sb / (n * m))

norm_x12 = [norm_x[i][0] * norm_x[i][1] for i in range(len(norm_x))]
norm_x13 = [norm_x[i][0] * norm_x[i][2] for i in range(len(norm_x))]
norm_x23 = [norm_x[i][1] * norm_x[i][2] for i in range(len(norm_x))]
norm_x123 = [norm_x[i][0] * norm_x[i][1] * norm_x[i][2] for i in
range(len(norm_x))]

natur_x12 = [[natur_x[i][0] * natur_x[i][1]] for i in range(len(natur_x))]
natur_x13 = [[natur_x[i][0] * natur_x[i][2]] for i in range(len(natur_x))]
natur_x23 = [[natur_x[i][1] * natur_x[i][2]] for i in range(len(natur_x))]
natur_x123 = [[natur_x[i][0] * natur_x[i][1] * natur_x[i][2]] for i in
range(len(natur_x))]

x1i = [norm_x[i][0] for i in range(n)]
x2i = [norm_x[i][1] for i in range(n)]
x3i = [norm_x[i][2] for i in range(n)]

list_b = [0] * n # b0, b1, b2, b3, b12, b13, b23, b123
list_b[0] = average(average_list_y)

```



```

list_b[1] = average([average_list_y[i] * x1i[i] for i in range(n)])
list_b[2] = average([average_list_y[i] * x2i[i] for i in range(n)])
list_b[3] = average([average_list_y[i] * x3i[i] for i in range(n)])
list_b[4] = average([average_list_y[i] * x1i[i] * x2i[i] for i in range(n)])
list_b[5] = average([average_list_y[i] * x1i[i] * x3i[i] for i in range(n)])
list_b[6] = average([average_list_y[i] * x2i[i] * x3i[i] for i in range(n)])
list_b[7] = average([average_list_y[i] * x1i[i] * x2i[i] * x3i[i] for i in
range(n)])

printed_matrixes()
print("\nPівніння\n" + regression_str.format(*map(round_to_2, list_b)))
cochrane_criteria(S_y)
student_criteria(S_y, d)
fisher_criteria(d)
#-----
m = 3
experiment(m)
+ {} * x1 + {} * x2 + {} * x3 + {} * x1x2 + {} * x1x3 + {} * x2x3 + {} *
x1x2x3'

def matrix_plan(m, ymin, ymax, n):
    return [[randint(ymin, ymax) for _ in range(m)] for _ in range(n)]

def multiplication(a, b):
    return a * b

def average(list):
    return sum(list) / len(list)

def round_to_2(number):
    return round(number, 2)

def dispersion(list_y, aver_list_y):
    return [round_to_2(average(list(map(lambda y: (y - aver_list_y[i]) ** 2,
list_y[i])))) for i in range(len(list_y)))]

def cochrane_criteria(S_y):
    global m
    print("\nКритерій Кохрена\n")
    Gp = max(S_y) / sum(S_y)
    q = 0.05
    q_ = q / f2
    chr = f.ppf(q=1 - q_, dfn=f1, dfd=(f2 - 1) * f1)
    Gt = chr / (chr + f2 - 1)
    print("Тест Кохрена: Gr = " + str(round(Gp, 3)))
    if Gp < Gt:
        print("Дисперсії однорідні з імовірністю 0.95")
        pass
    else:
        print("\nДисперсії неоднорідні.\nПовтор експерименту для m + 1\n")
        m = m + 1
        experiment(m)

def student_criteria(S_y, d):
    print("\nКритерій Ст'юдента\n")
    bettaList = [sum(S_y) * x0[0] / n,
                  average(list(map(multiplication, S_y, x1i))),
                  average(list(map(multiplication, S_y, x2i))),
                  average(list(map(multiplication, S_y, x3i))),

```

```

        average(list(map(multiplication, S_y, norm_x12))),
        average(list(map(multiplication, S_y, norm_x13))),
        average(list(map(multiplication, S_y, norm_x23))),
        average(list(map(multiplication, S_y, norm_x123))))]
bettaList = [round_to_2(i) for i in bettaList]

list_t = [bettaList[i] * S for i in range(n)]

for i in range(n):
    if list_t[i] < t.ppf(q=0.975, df=f3):
        list_b[i] = 0
        d -= 1
        print('Коефіцієнт b' + str(i) + ' незначимий, тому виключається
із рівняння регресії')
    print("\nСкореговане рівняння регресії:")
    print(regression_str.format(*map(round_to_2, list_b)))

def fisher_criteria(d):
    global m
    print("\nКритерій Фішера\n")
    f4 = n - d
    S_ad = (m * sum(
        [(list_b[0] + list_b[1] * x1i[i] + list_b[2] * x2i[i] + list_b[3] *
x3i[i] + list_b[4] * norm_x12[i] +
        list_b[5] * norm_x13[i] + list_b[6] * norm_x23[i] + list_b[7] *
norm_x123[i]
        - average_list_y[i]) ** 2 for i in range(n)]) / f4)
    Fp = S_ad / Sb

    if Fp > f.ppf(q=0.95, dfn=f4, dfd=f3): # перевірка критерію Фішера з
використанням scipy
        print('Математична модель неадекватна експериментальним даним на
рівні значимості 0.05.\nПовтор експерименту для m+1')
        m = m + 1
        experiment(m)
    else:
        print('Математична модель адекватна експериментальним даним на рівні
значущості 0.05')

def printed_matrixes():
    pt1 = prettytable.PrettyTable()
    pt2 = prettytable.PrettyTable()
    pt1.field_names = ["X0", "X1", "X2", "X3"] + ["X12", "X13", "X23",
"X123"] + ["Y" + str(x) for x in range(1, m + 1)] + ["Aver Y"] + ["S_y"]
    pt2.field_names = ["X0", "X1", "X2", "X3"] + ["X12", "X13", "X23",
"X123"] + ["Y" + str(x) for x in range(1, m + 1)] + ["Aver Y"] + ["S_y"]

    print("Матриця повного факторного експерименту з натуралізованими
значеннями:\n")
    pt1.add_rows([x0 + natur_x[i] + natur_x12[i] + natur_x13[i] +
natur_x23[i] + natur_x123[i] + matrix_y[i] + [average_list_y[i]] + [S_y[i]] for
i in range(n)])
    print(pt1)

    print("\nМатриця повного факторного експерименту з нормалізованими
значеннями:\n")
    pt2.add_rows([x0 + norm_x[i] + [norm_x12[i]] + [norm_x13[i]] +
[norm_x23[i]] + [norm_x123[i]] + matrix_y[i] + [average_list_y[i]] + [S_y[i]]
for i in range(n)])

```

```

        print(pt2)

m = m
n = 8

x_average_max = sum(max_x) / 3
x_average_min = sum(min_x) / 3

y_max = round(200 + x_average_max)
y_min = round(200 + x_average_min)

matrix_y = matrix_plan(m, y_min, y_max, n)
average_list_y = [round(average(matrix_y[i]), 2) for i in
range(len(matrix_y))]

S_y = dispersion(matrix_y, average_list_y)

f1 = m - 1
f2 = n
f3 = f1 * f2
d = 4

Sb = sum(S_y) / n
S = math.sqrt(Sb / (n * m))

norm_x12 = [norm_x[i][0] * norm_x[i][1] for i in range(len(norm_x))]
norm_x13 = [norm_x[i][0] * norm_x[i][2] for i in range(len(norm_x))]
norm_x23 = [norm_x[i][1] * norm_x[i][2] for i in range(len(norm_x))]
norm_x123 = [norm_x[i][0] * norm_x[i][1] * norm_x[i][2] for i in
range(len(norm_x))]

natur_x12 = [[natur_x[i][0] * natur_x[i][1]] for i in range(len(natur_x))]
natur_x13 = [[natur_x[i][0] * natur_x[i][2]] for i in range(len(natur_x))]
natur_x23 = [[natur_x[i][1] * natur_x[i][2]] for i in range(len(natur_x))]
natur_x123 = [[natur_x[i][0] * natur_x[i][1] * natur_x[i][2]] for i in
range(len(natur_x))]

x1i = [norm_x[i][0] for i in range(n)]
x2i = [norm_x[i][1] for i in range(n)]
x3i = [norm_x[i][2] for i in range(n)]

list_b = [0] * n # b0, b1, b2, b3, b12, b13, b23, b123
list_b[0] = average(average_list_y)
list_b[1] = average([average_list_y[i] * x1i[i] for i in range(n)])
list_b[2] = average([average_list_y[i] * x2i[i] for i in range(n)])
list_b[3] = average([average_list_y[i] * x3i[i] for i in range(n)])
list_b[4] = average([average_list_y[i] * x1i[i] * x2i[i] for i in range(n)])
list_b[5] = average([average_list_y[i] * x1i[i] * x3i[i] for i in range(n)])
list_b[6] = average([average_list_y[i] * x2i[i] * x3i[i] for i in range(n)])
list_b[7] = average([average_list_y[i] * x1i[i] * x2i[i] * x3i[i] for i in
range(n)])

printed_matrixes()
print("\nPівніжння\n" + regression_str.format(*map(round_to_2, list_b)))
cochrane_criteria(S_y)
student_criteria(S_y, d)
fisher_criteria(d)
#-----

```

```
m = 3  
experiment(m)
```